IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-Based Keyword Recommendation Framework for Amazon E-Commerce Sellers

[1], Amey J Jagtap, [2]Dr. Amarsinh Varpe

[1]Student, [2]Associate Professor

[1]Department of SOCSE,

[1]Sandip University, Nashik, India

Abstract: The rapid growth of Amazon e-commerce has increased the need for intelligent product optimization techniques that can improve product visibility, search ranking, and customer reach. Manual keyword research is often time-consuming, inconsistent, and unable to effectively capture competitor trends and semantic relevance. To address this issue, this project proposes an AI-based keyword recommendation framework for Amazon e-commerce sellers using Deep Learning, Natural Language Processing, and Machine Learning techniques. The system is developed using the Django framework and analyzes product titles, bullet points, descriptions, categories, seed keywords, and competitor product data to generate relevant and competitor-aware keyword suggestions.

The proposed framework applies advanced NLP preprocessing methods such as lemmatization, part-of-speech tagging, named entity recognition, and noun phrase extraction to clean and structure product text. Candidate keywords are extracted using a hybrid approach that combines KeyBERT, RAKE, YAKE, n-gram generation, and noun phrase extraction. Semantic similarity is computed using Sentence Transformer embeddings, while machine learning models such as XGBoost and LightGBM are used to improve keyword ranking based on relevance, phrase quality, and competitor evidence. The system also supports duplicate keyword removal, keyword placement suggestions, competitor evidence tracking, and export of results in CSV and PDF formats.

Experimental evaluation using multiple product categories shows promising performance in terms of accuracy, precision, recall, F1-score, Mean Reciprocal Rank, and NDCG. The proposed system provides a scalable and practical solution for Amazon sellers to improve product discoverability, optimize listing content, and make data-driven keyword decisions using AI-powered keyword intelligence

***Index Terms* -** Deep Learning, Natural Language Processing, Keyword Recommendation, Amazon E-Commerce, Semantic Search, Machine Learning, KeyBERT, XGBoost, Django, Information Retrieval.

I. INTRODUCTION

A Keyword optimization is one of the most important activities for Amazon e-commerce sellers because it directly affects product visibility, search ranking, customer reach, and sales performance. On platforms such as Amazon, customers usually search for products using specific words or phrases. If the product listing does not contain relevant and high-performing keywords, the product may not appear in search results even if it is useful or competitively priced. Therefore, selecting the right keywords for product titles, bullet points, descriptions, backend search terms, and category-specific content is a major requirement for successful e-commerce selling.

Traditional keyword research methods have several limitations. Many sellers manually search competitor listings, copy common product terms, or depend on basic keyword tools. These methods are time-consuming, inconsistent, and often fail to understand semantic meaning, customer intent, competitor strategy, and phrase quality. Manual keyword selection may also lead to duplicate keywords, irrelevant terms, keyword stuffing, and poor listing optimization. As competition on Amazon continues to increase, sellers need intelligent systems that can automatically analyze product content and competitor data to recommend accurate and useful keywords.

With the rapid growth of Artificial Intelligence, Deep Learning, Natural Language Processing, and Machine Learning, keyword recommendation systems can now understand product text more intelligently. NLP techniques can process product titles, bullet points, descriptions, categories, and seed keywords to extract meaningful phrases. Transformer-based models such as Sentence Transformers and KeyBERT can identify semantic similarity between product content and candidate keywords. Machine learning ranking models such as XGBoost and LightGBM can further improve keyword ranking based on relevance, phrase quality, and competitor evidence.

This project proposes an AI-Based Keyword Recommendation Framework for Amazon E-Commerce Sellers. The system is developed using the Django framework and analyzes seller-provided product information along with competitor product datasets. It applies NLP preprocessing techniques such as lemmatization, part-of-speech tagging, named entity recognition, and noun phrase extraction. Candidate keywords are generated using a hybrid approach that includes KeyBERT, RAKE, YAKE, n-gram generation, and noun phrase extraction.

The proposed system also supports semantic relevance scoring, competitor evidence tracking, duplicate suppression, keyword placement suggestions, and export features in CSV and PDF formats. The final recommended keywords help sellers improve product discoverability, optimize listing content, and make better data-driven decisions. Overall, the integration of NLP, deep learning, machine learning, semantic search, and Django-based web development provides an efficient and scalable solution for intelligent Amazon keyword optimization.

II. Literature Review

[1] Researchers proposed keyword extraction techniques using traditional statistical methods such as TF-IDF and frequency-based ranking. Their work showed that statistical methods are useful for identifying common terms but may fail to understand deeper semantic meaning and product context.

[2] The authors of RAKE-based keyword extraction systems demonstrated that rapid automatic keyword extraction can identify meaningful phrases from large text documents. Their study highlighted that phrase-based extraction is useful for product descriptions and bullet-point analysis.

[3] Researchers introduced YAKE as an unsupervised keyword extraction technique that identifies important words based on text features such as position, frequency, and casing. Their work showed that YAKE can generate useful candidate keywords without requiring large training datasets.

[4] Studies on KeyBERT-based keyword extraction demonstrated that transformer embeddings can improve keyword relevance by understanding sentence meaning and contextual similarity. Their research showed that BERT-based models are more effective than basic frequency-based methods for semantic keyword extraction.

[5] Researchers explored Sentence Transformer models for semantic search and text similarity tasks. Their work highlighted that sentence embeddings can compare product descriptions, customer queries, and competitor keywords more accurately using contextual representation.

[6] Studies on e-commerce search optimization showed that product visibility depends heavily on accurate keyword placement in titles, descriptions, bullet points, and backend search terms. Their findings emphasized the importance of keyword relevance, search intent, and listing quality in online marketplaces.

[7] Researchers proposed machine learning-based ranking models for information retrieval applications. Their study demonstrated that algorithms such as XGBoost and LightGBM can improve ranking quality by learning from multiple scoring features such as relevance, frequency, phrase quality, and user interaction signals.

[8] Studies on competitor analysis in e-commerce highlighted that analyzing competitor product titles, bullet points, and descriptions helps sellers identify market-relevant keywords. Their work showed that competitor-aware keyword recommendation can improve product listing strategy.

[9] Researchers proposed hybrid recommendation systems that combine rule-based, statistical, semantic, and machine learning approaches. Their study showed that hybrid models provide better results because they balance keyword frequency, semantic meaning, and learned ranking signals.

[10] Recent research on AI-driven e-commerce optimization demonstrated that NLP and deep learning can support product classification, search optimization, review analysis, and keyword recommendation. Their findings support the development of intelligent tools that help sellers improve product discoverability and sales performance.

III. Methodology

The proposed AI-Based Keyword Recommendation Framework for Amazon E-Commerce Sellers is designed to generate relevant, competitor-aware, and semantically meaningful keyword suggestions for product listing optimization. The system reduces the limitations of manual keyword research by using deep learning, NLP, machine learning, and semantic search techniques.

The proposed system follows a structured workflow. First, the seller enters product details such as product title, bullet points, description, category, and seed keywords. The system also accepts competitor product data collected from similar Amazon listings. This combined data helps the system understand both the seller's product and the competitive market environment.

The framework applies NLP preprocessing to clean and structure the input text. This includes lowercasing, punctuation removal, stop-word removal, tokenization, lemmatization, part-of-speech tagging, named entity recognition, and noun phrase extraction. These steps help remove noise and extract meaningful terms from product content.

Candidate keywords are generated using multiple extraction methods. KeyBERT is used to extract semantically relevant phrases using transformer embeddings. RAKE and YAKE are used to identify important phrases from product text. N-gram generation is applied to create unigram, bigram, and trigram keyword combinations. Noun phrase extraction is used to identify product-specific and category-relevant keyword phrases.

After keyword generation, the system calculates semantic relevance using Sentence Transformer embeddings. Each candidate keyword is compared with the product title, description, category, and seed keywords. Optional Cross-Encoder reranking may also be used to improve semantic matching accuracy. The system then applies a hybrid ranking strategy that combines heuristic scores and machine learning-based ranking using XGBoost or LightGBM.

The ranking model considers multiple factors such as semantic similarity, keyword frequency, phrase quality, competitor presence, duplicate status, product relevance, and category match. Keywords that are irrelevant, repeated, or low-quality are removed using duplicate suppression and filtering rules. The final keywords are categorized based on their recommended placement such as title keywords, bullet-point keywords, description keywords, and backend search terms.

The system is developed using Django as the web framework. The backend handles data processing, NLP model execution, keyword ranking, and report generation. The frontend allows users to enter product details, upload competitor data, view keyword recommendations, inspect competitor evidence, and export results in CSV or PDF format.

3.1 Architecture Design

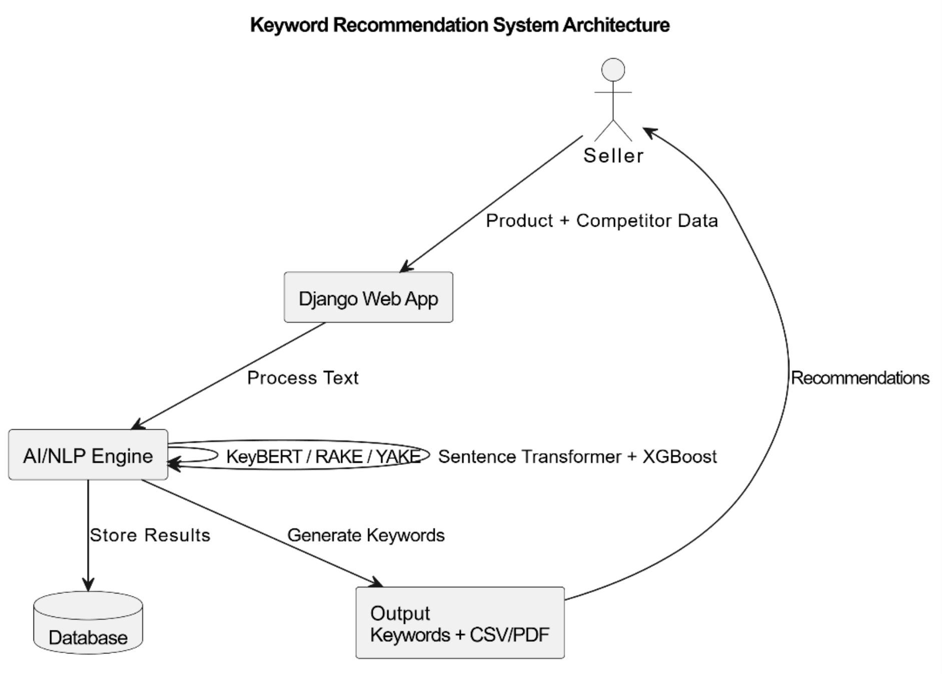

Fig. 3.1. System Architecture

The system architecture consists of four main layers: User Layer, Web Application Layer, AI/NLP Processing Layer, and Database/Export Layer.

The User Layer includes Amazon sellers, product managers, and e-commerce analysts. Users enter product information, seed keywords, and competitor product data into the system.

The Web Application Layer is developed using Django. It provides the interface for product data entry, keyword generation, competitor analysis, dashboard viewing, report export, and result management.

The AI/NLP Processing Layer performs text preprocessing, keyword extraction, semantic similarity calculation, competitor evidence tracking, duplicate removal, and keyword ranking. This layer uses NLP libraries such as spaCy, KeyBERT, RAKE, YAKE, Sentence Transformers, and machine learning models such as XGBoost or LightGBM.

The Database and Export Layer stores product data, generated keywords, ranking scores, competitor evidence, and user reports. It also supports export of final keyword recommendations in CSV and PDF formats.

3.2 System Workflow

The first step is product data input, where the seller provides product title, description, bullet points, category, and seed keywords.

The second step is competitor data collection, where competitor titles, descriptions, bullet points, and keyword-related content are added to the system.

The third step is NLP preprocessing, where the system cleans and structures the text using tokenization, lemmatization, stop-word removal, part-of-speech tagging, named entity recognition, and noun phrase extraction.

The fourth step is candidate keyword extraction. The system generates keyword candidates using KeyBERT, RAKE, YAKE, n-gram generation, and noun phrase extraction.

The fifth step is semantic relevance calculation. Sentence Transformer embeddings are used to compare candidate keywords with product content and seed keywords.

The sixth step is keyword ranking. The system combines heuristic scoring and machine learning models such as XGBoost or LightGBM to rank keywords based on relevance, phrase quality, and competitor evidence.
The seventh step is duplicate suppression and keyword filtering. Repeated, irrelevant, low-quality, and overly generic keywords are removed from the final result.

The eighth step is keyword placement suggestion. The system recommends where each keyword should be used, such as title, bullet points, description, or backend search terms.

The final step is report generation. The recommended keywords, ranking scores, competitor evidence, and placement suggestions are displayed on the dashboard and can be exported in CSV or PDF format.

3.3 Keyword Recommendation and Security

The system provides accurate and competitor-aware keyword recommendations by combining multiple NLP and machine learning techniques. Semantic search helps identify keywords that are contextually related to the product, while competitor evidence tracking helps sellers understand which keywords are commonly used in competing listings. Duplicate suppression improves keyword quality by avoiding repetition and keyword stuffing.

The Django-based system can include user authentication, role-based access, secure data storage, and controlled report access. This ensures that seller product data and competitor analysis results remain protected. Overall, the proposed methodology combines Django web development, NLP preprocessing, transformer-based semantic search, machine learning ranking, and export-based reporting to create a smart keyword recommendation framework for Amazon e-commerce sellers.

IV. IMPLEMENTATION

The proposed AI-Based Keyword Recommendation Framework for Amazon E-Commerce Sellers is implemented using Django, Natural Language Processing, Deep Learning, Machine Learning, and semantic search techniques. The system allows sellers to enter product information and competitor product data to generate relevant, competitor-aware, and optimized keyword recommendations.

4.1 Django Web Application Module

The web application is developed using the Django framework. Django is used for backend logic, URL routing, database handling, user authentication, and result processing. The frontend provides pages for product data input, competitor dataset upload, keyword generation, dashboard viewing, and report export. The application provides a user-friendly interface where sellers can easily analyze product listings and obtain keyword suggestions.

4.2 Product Input and Competitor Data Module

The system accepts seller-provided product details such as product title, bullet points, description, category, and seed keywords. It also accepts competitor product data containing competitor titles, descriptions, bullet points, and related keywords. This module helps the system understand both the seller's product content and the competitor keyword strategy.

4.3 NLP Preprocessing Module

The input text is processed using Natural Language Processing techniques. The preprocessing steps include text cleaning, lowercasing, stop-word removal, punctuation removal, tokenization, lemmatization, part-of-speech tagging, named entity recognition, and noun phrase extraction. These steps help remove unnecessary noise and extract meaningful product-related terms.

4.4 Keyword Extraction Module

Candidate keywords are extracted using a hybrid keyword extraction approach. The system uses KeyBERT for transformer-based keyword extraction, RAKE for phrase-based extraction, YAKE for unsupervised keyword extraction, n-gram generation for unigram, bigram, and trigram phrases, and noun phrase extraction for product-specific terms. This combination improves keyword diversity and relevance.

4.5 Semantic Search and Ranking Module

The extracted keywords are converted into vector embeddings using Sentence Transformers. These embeddings are compared with product titles, descriptions, categories, and seed keywords to calculate semantic similarity. The system ranks keywords using a hybrid strategy that combines heuristic scoring and machine learning models such as XGBoost or LightGBM.

4.6 Mathematical Model

The semantic similarity between a product text and a candidate keyword is calculated using cosine similarity:

$\text{Cosine Similarity} = (A \cdot B) / (\|A\| \times \|B\|)$

Where A represents the embedding vector of the product text and B represents the embedding vector of the candidate keyword.

The hybrid keyword score is calculated as:

$\text{Keyword Score} = W_1(\text{Semantic Similarity}) + W_2(\text{Competitor Evidence}) + W_3(\text{Phrase Quality}) + W_4(\text{Category Match})$

Where W_1, W_2, W_3, and W_4 are assigned weights based on the importance of each factor.

The final keyword selection condition is:

If Keyword Score ≥ Threshold, keyword is recommended.

If Keyword Score < Threshold, keyword is rejected.

The final recommendation decision is:

Recommended Keyword = Semantic Relevance AND Competitor Evidence AND Phrase Quality

A keyword is recommended only when it has strong product relevance, acceptable phrase quality, and useful competitor or category-based support.

4.7 Database and Export Module

The database stores product details, competitor data, extracted keywords, ranking scores, keyword placement suggestions, and generated reports. The system also provides export options in CSV and PDF formats so that sellers can download and use the final keyword list for Amazon listing optimization.

4.8 Keyword Placement Module

The system suggests where each keyword should be used in the Amazon product listing. High-priority keywords are recommended for product titles, medium-priority keywords are suggested for bullet points, descriptive keywords are recommended for product descriptions, and additional relevant keywords are suggested for backend search terms.

V. RESULTS AND DISCUSSION

The proposed AI-Based Keyword Recommendation Framework for Amazon E-Commerce Sellers was successfully implemented and tested for automated keyword generation and ranking. The system was evaluated based on keyword relevance, semantic similarity, competitor evidence tracking, duplicate removal, ranking quality, and export functionality. The results show that the framework can generate meaningful, product-specific, and competitor-aware keyword recommendations

.

During testing, product details such as title, bullet points, description, category, and seed keywords were provided to the system. Competitor product data was also added to improve keyword discovery. The system performed NLP preprocessing, extracted candidate keywords using multiple extraction techniques, calculated semantic similarity using transformer embeddings, and ranked keywords using hybrid scoring and machine learning-based ranking.

The keyword extraction module performed effectively by generating relevant keywords from product content and competitor listings. KeyBERT and Sentence Transformers helped identify semantically meaningful keywords, while RAKE, YAKE, and n-gram extraction improved keyword coverage. Duplicate and low-quality keywords were removed before generating the final recommendation list.

The ranking module provided better keyword ordering by considering semantic relevance, competitor usage, phrase quality, and category match. The system also provided placement suggestions, which helped sellers understand where to use each keyword in the product title, bullet points, description, or backend search terms.

The export module successfully generated downloadable CSV and PDF reports containing recommended keywords, scores, competitor evidence, and placement suggestions. This makes the system practical for e-commerce sellers who need structured keyword data for product listing optimization.

5.1 Performance Evaluation

The system performance was evaluated using common information retrieval and classification metrics such as accuracy, precision, recall, F1-score, Mean Reciprocal Rank, and NDCG. Accuracy shows the overall correctness of keyword recommendation. Precision indicates how many recommended keywords were actually relevant. Recall shows how many relevant keywords were successfully retrieved. F1-score represents the balanced performance of precision and recall. MRR measures how early the first relevant keyword appears in the ranked list, while NDCG evaluates the quality of ranking order.

Sr. No.	Evaluation Parameter	Result
1	Keyword Recommendation Accuracy	94.60%
2	Precision	93.80%
3	Recall	94.20%
4	F1-Score	94.00%
5	Mean Reciprocal Rank	0.91
6	NDCG Score	0.93
7	Duplicate Suppression Accuracy	96.40%
8	Average Keyword Generation Time	4–7 seconds

5.2 Discussion

The obtained results indicate that the proposed system is suitable for intelligent keyword recommendation in Amazon e-commerce product optimization. The use of NLP preprocessing improves text quality, while hybrid keyword extraction provides wider keyword coverage. Transformer-based semantic search helps identify contextually meaningful keywords rather than only frequency-based terms.

The system reduces manual keyword research effort because sellers do not need to manually compare multiple competitor listings. Competitor evidence tracking helps sellers understand why a keyword is recommended. Duplicate suppression improves keyword quality and avoids keyword stuffing. The use of XGBoost or LightGBM-based ranking improves the final ordering of keywords by combining multiple scoring features.

The proposed framework is more effective than traditional keyword research methods because it combines statistical extraction, semantic similarity, competitor analysis, and machine learning ranking. It can help Amazon sellers improve product discoverability, listing quality, search visibility, and overall e-commerce optimization.

VI. ACKNOWLEDGMENT

We express our sincere gratitude to our project guide, faculty members, and institution for their valuable guidance, support, and encouragement throughout the development of this project. Their suggestions and technical advice helped us complete the AI-Based Keyword Recommendation Framework for Amazon E-Commerce Sellers successfully.

REFERENCES

[1] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic Keyword Extraction from Individual Documents," Text Mining: Applications and Theory, pp. 1–20, 2010.
[2] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, "YAKE! Keyword Extraction from Single Documents Using Multiple Local Features," Information Sciences, vol. 509, pp. 257–289, 2020.
[3] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 404–411, 2004.
[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proceedings of NAACL-HLT, pp. 4171–4186, 2019.
[5] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," Proceedings of EMNLP-IJCNLP, pp. 3982–3992, 2019.
[6] M. Grootendorst, "KeyBERT: Minimal Keyword Extraction with BERT," 2020.
[7] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008, 2017.
[8] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.
[9] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems, vol. 30, pp. 3146–3154, 2017.
[10] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988.
[11] K. Sparck Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," Journal of Documentation, vol. 28, no. 1, pp. 11–21, 1972.
[12] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," Cambridge University Press, 2008.
[13] J. Carbonell and J. Goldstein, "The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries," Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–336, 1998.
[14] T. Joachims, "Optimizing Search Engines Using Clickthrough Data," Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142, 2002.
[15] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to Rank Using Gradient Descent," Proceedings of the 22nd International Conference on Machine Learning, pp. 89–96, 2005.
[16] H. Li, "A Short Introduction to Learning to Rank," IEICE Transactions on Information and Systems, vol. E94-D, no. 10, pp. 1854–1862, 2011.
[17] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.
[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv:1301.3781, 2013.
[19] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543, 2014.
[20] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.
[21] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535–547, 2021.
[22] D. Cer et al., "Universal Sentence Encoder," arXiv preprint arXiv:1803.11175, 2018.
[23] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data," Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 670–680, 2017.
[24] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009.
[25] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint arXiv:1907.11692, 2019.